1. **Write two to three sentences on why Django is so popular among web developers.**

Django is highly popular among web developers due to its "batteries-included" approach, offering a comprehensive standard library that helps in rapid development and deployment of web applications. Furthermore, its robust security features and scalability make it an attractive choice for developers looking to build secure, maintainable, and scalable web applications efficiently.

1. **After some research, list five large companies that use Django. Specify what the company’s product or service is and what they use Django for.**

**Instagram**: A social networking service focused on photo and video sharing, Instagram uses Django to handle massive amounts of data and user interactions, leveraging Django’s scalability.

**Spotify**: A digital music service offering streaming of a vast music library, Spotify employs Django to manage its backend services and user experience with efficiency and speed.

**Mozilla**: Known for the Firefox browser, Mozilla uses Django for the development of many of its web applications, valuing Django's security features and scalability.

**Dropbox**: A leading cloud storage service, Dropbox uses Django for handling web applications that manage file storage, sharing, and synchronization services across devices.

**Pinterest**: A visual discovery engine and social networking platform, Pinterest utilizes Django to manage its vast database of pins and user interactions, capitalizing on Django’s ability to efficiently handle high volumes of data.

1. **For each of the following scenarios, explain if you would use Django (and why or why not):**
   * **You need to develop a web application with multiple users.**

Yes, you would use Django because it provides built-in support for handling user authentication, sessions, and permissions, which are essential for managing multiple users.

* + **You need fast deployment and the ability to make changes as you proceed.**

Yes, Django is suitable because its "batteries-included" approach and automatic admin interface can speed up development and allow for rapid changes.

* + **You need to build a very basic application, which doesn’t require any database access or file operations.**

No, you might not choose Django for this scenario as it might be overkill; a simpler framework or even static site generators could be more appropriate.

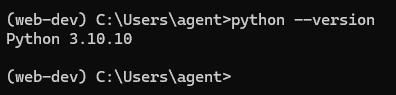
* + **You want to build an application from scratch and want a lot of control over how it works.**

While Django offers flexibility and customization, it may not be the optimal choice for projects requiring absolute control over every detail, due to its "batteries-included" approach and predefined structure. Frameworks like Flask in the Python ecosystem, or Express in Node.js, offer more flexibility at the cost of requiring more decisions and implementations to be made from scratch.

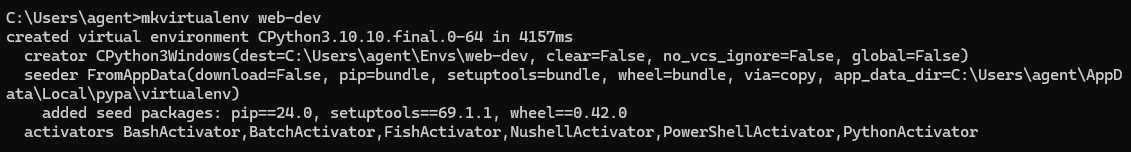
* + **You’re about to start working on a big project and are afraid of getting stuck and needing additional support.**

Yes, Django is a good choice because it has a large and active community, providing extensive documentation, resources, and forums for getting help and support, reducing the risk of getting stuck.

Python version:



Creating a virtual environment:



Activating the virtual environment, then installing Django and checking that it’s installed

